COCOLOG: a conditional observer and controller logic for finite machines

Peter E. Caines, Suning Wang

INRIA

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

1 9 9 2

25ème anniversaire

N° 1714

Programme 5
Traitement du Signal,
Automatique et Productique

COCOLOG : A CONDITIONAL OBSERVER AND CONTROLLER LOGIC FOR FINITE MACHINES

Peter E. CAINES
Suning WANG

Juin 1992

RR-1714

COCOLOG: A Conditional Observer and Controller Logic for Finite Machines

P.E. Caines [1]
Dept. of Electrical Engineering, McGill University,
Montreal, P.Q. H3A 2A7, Canada,
and
Canadian Institute for Advanced Research

S. Wang [2]
SPAR Aerospace Ltd, 21025 Trans-Canada Highway
St-Anne-de-Bellevue, PQ, Canada

[1]Work supported by NSERC Grant A1329 and INRIA - Sophia Antipolis, France.
[2]Work performed at McGill University.

La Logique Conditionelle "COCOLOG" pour l'Observation et la Commande des Machines à Etat Fini

Peter E. Caines Suning Wang

Résumé

On considère des problèmes d'observation et de commande des machines à états finis (à entrée–état–sortie) ; ces problèmes sont formulés par un arbre de théories de logique du premier ordre. Etant donné une machine quelconque M, on y associe un ensemble de langages du premier ordre qui facilite la description de l'évolution contrôlée et de l'estimation de l'état de M ; de plus, les théories du premier ordre possèdent des "axiomes de commande conditionelle" qui prescrivent des actions de commande en boucle fermée lorsque des conditions (qui dépendent de l'histoire du système) sont satisfaites. En particulier, ces axiomes peuvent contenir des groupes de commandes qui font passer le système d'un état (ou d'un état estimé) à un état cible, si on peut démontrer l'existence d'une telle séquence. La théorie initiale est la théorie générale de M et puis à chaque instant suivant, on accepte des nouvelles observations (des entrées–sorties du système) comme axiomes nouveaux ; ceux–ci s'accumulent pour conduire les nouvelles théories. On appelle une telle collection de théories, pour n'importe quel système, un COCOLOG (acronyme pour les mots anglais "conditional observer and controller logic"). Dans cet article on donne une sémantique pour chaque COCOLOG fondée sur les segments initiaux (partiellements ordonnés) des séquences entrées–sorties d'une machine. Des règles "hors logique" établissent la correspondance entre les théories d'une COCOLOG. Dans cet article on démontre la complétude et la consistance de chaque théorie d'une COCOLOG et la décidabilité est aussi établie – grâce à la propriété qu'il n'existe qu'un modèle de chaque théorie. Des exemples du fonctionement des contrôleurs COCOLOG sur des machines finies simples sont présentés.

Abstract

The problem of observation and control for partially observed input-state-output machines is formulated in terms of a tree of first order logical theories. A set of first order languages for the description of the controlled evolution and state estimation of any given machine $\mathcal{M}$ is specified; further, conditional control axioms are formulated so that closed loop control actions occur when specified past observation dependent conditions are fulfilled. In particular, conditional control axioms may include commands that steer the system state from a current partially observed state (estimate) to a target state, if such a sequence of controls can be proven to exist. Starting from a general theory of $\mathcal{M}$ at the initial instant, observations on the input-output behaviour of the system at any later instant are accepted by the system as new axioms; these are then used together with the previously generated theory to generate the current theory. We use the acronym COCOLOG to denote the family of first order *conditional observer and controller logics* for any given input-state-output system. A semantics is supplied for each COCOLOG in terms of interpretations of controlled transitions on a tree indexed by the possible sequences of input-output observations. Extra-logical rules relating members of the family of theories of a COCOLOG are then presented in the form of meta-level axioms and inference rules. In this paper consistency and completeness of the first order theories in a COCOLOG family are established, decidability is obtained using a unique model property and examples of the operation of a COCOLOG logic control system are given.

1 Introduction

In this paper we introduce certain partially ordered sets of first order logical theories which we call *conditional observer and controller logics*, or *COCOLOGs* for short. A COCOLOG provides a logical system for (i) describing and reasoning about the state estimation and control of a given finite input-state-output machine $\mathcal{M}$, and (ii) acting upon $\mathcal{M}$ via a closed loop logic regulator $\mathcal{R}$ carrying the corresponding COCOLOG. (By abuse of language, we shall refer to all such COCOLOG-structures by the generic term COCOLOG.)

A particular subset of the axioms in each of the constituent logical theories of a COCOLOG is called the set of *conditional control axioms (CCAs)*; these are formulated so that certain control actions are specified at an instant k when certain past measurable (i.e. past observations dependent) conditions C_k are fulfilled. In COCOLOG this translates precisely into the existence of a proof, in the corresponding first order logical theory, of a predicate describing the conditions C_k. Conditional control statements may include, for example, control commands that will steer the current system state, or its estimate, to a given target state x^T; such commands would be implemented whenever a sequence of controls achieving this objective can be proven to exist, with the uniqueness of the selected control ensured via a prescribed arrangement of the CCAs.

The conceptualization of a feedback regulator system adopted in this paper is qualitatively different from the usual notion of a feedback system. In the customary formulation, a classical feedback regulator R is an input-output dynamical system whose inputs are typically the measured outputs of the controlled system Σ and whose outputs are the controlled inputs to Σ. Hence the system and the regulator are objects of the same type, namely input (-state)- output dynamical systems. However, in our formulation, when Σ is in a feedback loop with a logico-linguistic regulator (henceforth simply a logic regulator) $\mathcal{R}$ the situation is quite different and we now give a sketch of the operation of the system $(\Sigma, \mathcal{R})$.

At each discrete time instant k, the output y_k of the system Σ, taking the value $y^j \epsilon Y$, is mapped extra-logically into a predicate which is accepted by the regulator $\mathcal{R}$. In the present case, $\mathcal{R}$ is conceived of as a *dynamical logical system* mapping theories to theories (see Caines, Greiner, Wang [1988], [1991]) and emitting outputs via a second extra-logical map. Let the theory carried by $\mathcal{R}$ at the instant $k-1$ be denoted $Th(o_1^{k-1})$, where o_1^{k-1} denotes the sequence of observations over $[1, k-1]$. At the instant k the equality predicate relating the constant $Y(k)$ at k to the observed quantity y^j is accepted as new information into the theory $Th(o_1^{k-1})$. By this we mean that the equality predicate is taken as a new axiom to be added to $Th(o_1^{k-1})$. In addition, the conditional control axioms CCA_k and the state estimation axioms indexed by k, are also accepted as new axioms. The theory carried by $\mathcal{R}$ is then transformed into the deductive closure of (i) $Th(o_1^{k-1})$, (ii) the state estimation axioms and (iii) the axioms CCA_k, and this is relabeled as $Th(o_1^k)$. By their design, the conditional control axioms and the state estimation axioms CCA_k yield, within $Th(o_1^k)$, unique, deducible, values $U(k)$ for the input to Σ at k. The predicates defining these values are then mapped by the second extra-logical transformation referred to above into quantities which form the inputs to Σ at the instant k. This completes the dynamics-to-logical theory cycle $\Sigma \rightarrow \mathcal{R} \rightarrow \Sigma$. The system Σ now performs another dynamical evolution

step to generate the observed output y_{k+1} at the instant $k+1$ and the cycle repeats.

The process above is initiated with the system Σ in its initial state x_0 and the regulator $\mathcal{R}$ carrying only $Th_0 \triangleq Th(o_1^o)$, where $Th(o_1^o)$ consists of the deductive closure of the dynamical axioms of $\mathcal{M}$ (i.e. those describing the state transition and output maps), together with the logical axioms, the axioms for equality and the axioms for the reachability predicates.

(It is evident that the $\Sigma \rightarrow \mathcal{R} \rightarrow \Sigma$ feedback loop may be generalized to a loop $\mathcal{L} \rightarrow \mathcal{R} \rightarrow \mathcal{L}$, where $\mathcal{L}$ is itself a dynamical logical system, but the investigation of this is left for future work.)

The exposition in this paper is in terms of finite state machines solely to establish the theory of COCOLOG in its simplest context. There is no obstruction, in principle, to extending the theory to machines in continuous time, extended state machines and the automata of Ramadge-Wonham DES theory (see, for instance, Ramadge and Wonham [1987,1989], Wonham and Ramadge [1987] and Lin and Wonham [1988]) ; such extensions form a part of our program.

The development of COCOLOG for dynamical control systems has a two-fold motivation: first, the hierarchical nature of contemporary computer controlled systems may be better understood and enhanced by a study of regulator systems conceptualized at the logico-linguistic level. A notable example in this context is the capacity of reasoning systems to accept and operate on existential assertions, something a classical dynamical regulator is, of course, incapable of doing. Second, a control objective in COCOLOG, such as steering the system state to some state x^T, may be modified at any instant in the controlled machine's operation by conditions which are expressed via conjunctions and disjunctions of predicates; such conditions will be accepted by a COCOLOG regulator as new CCAs and on the basis of these new control laws will be deduced. By their nature, conventional dynamical regulators cannot easily accept significantly modified objectives, but must be redesigned to fit a new task. (We note this is in contradistinction to the ability of conventional regulators with fixed control objectives to adapt to changing system dynamics.) We conjecture that, over certain non-trivial classes of control problems, the computational cost of an automatic (adaptive) deduction of a new control law as a result of changes of control objectives (and possibly dynamics) will be smaller than the cost of pre-computing the appropriate laws for all such possibilities. It would also appear that information concerning system performance and objectives which involves flexible combinations of rules, necessary conditions and sets of alternatives is best expressed logico-linguistically and hence a logic based controller is most suitable for operating in this domain.

Previous work on the formulation of the theory presented here and its ramifications has appeared in the papers Caines, Greiner, Wang [1991], Caines, Wang, Greiner [1988], Caines and Wang [1989 a,b], Wang and Caines [1991], and, in particular, Caines and Wang [1990] and the thesis of Wang [1991] on which the exposition in this paper is in part based.

Earlier works which are analogous to, but different from, that presented here are the situated automaton work of Rosenschein and Kaebling [1987] and the line of research of Thistle and Wonham [1986], Ostroff [1987, 1989a, 1989b] and Ostroff and Wonham [1985, 1989]. In the latter two sets of work a fully elaborated temporal logic framework is presented to verify the correctness of feedback control algorithms for extended state machines. More

recently, Kohn [1988, 1991] has devised a formulation of the logic control problem in which equational axiom systems describe the dynamical properties of continuous time systems and the declarative language of the system expresses optimization goals and constraints. Automatic automata based inference procedures then create what is called a declarative control architecture.

We conclude this introduction with a brief remark about computational implementation. In its most direct implementation, a COCOLOG controller requires the real time implementation of automatic theorem proving (ATP) programs. The status of automatic theorem proving might suggest that this would be a formidable task. However, the restrictive nature of the system dynamical axioms and the nature of certain restricted classes of CCAs provide an opportunity to increase the efficiency of standard ATP programs. This has been initiated in Wang and Caines [1991], where a technique called Function Evaluation (FE) resolution is introduced. This essentially permits one to exploit the purely dynamical properties of the system via certain syntactic programs and tables that work in parallel with a given ATP program. Initial experiments have been conducted in which FE-resolution is implemented (using software developed by Q-X. Yu) in conjunction with the ATP software (GTP) of Newborn [1987]. The results are encouraging in that the FE-GTP software demonstrates simple theorems about non-trivial machines that defeat conventional ATP software (in this case GTP without the FE extension).

2 Finite State Machines

In this section we formally introduce our finite machine set-up and define just those observation and control notions which will be required in subsequent sections of this paper.

Definition 2.1. A *(partially observed) finite (input-state-output) machine is a quintuple* $\mathcal{M} = (X, U, Y, \Phi, \eta)$ where X is a (finite) set of *states*, U it is a (finite) set of *inputs*, Y is a (finite) set of *outputs*, $\Phi : X \times U \rightarrow X$ is a *transition function*, $\eta : X \rightarrow Y$ is an *output function.*

□

Concerning notation, we shall sometimes write u_i^n for the $(n - i + 1)$–element sequence $[u_i, u_{i+1}, u_{i+2}, \cdots, u_n]$, where $u_j \in U$ denotes the input at the time instance $j \in \mathcal{Z}_+$ (and where u_j is identified with $[u_j]$) and ϕ^u will denote the empty input string; the same notation will also be used for output sequences.

The dynamical evolution of a finite machine $\mathcal{M} = (X, U, Y, \Phi, \eta)$ can be displayed by taking U^* to be the set of all finite sequence of inputs and by extending $\Phi : X \times U \rightarrow X$ to $\Phi : X \times U^* \rightarrow X$, where for all $i, n \in \mathbb{N}_+$, for all $u_i^n \in U^*$ and for all $x \in X$, Φ is recursively defined by:

$$\begin{aligned} \Phi(x, \phi^u) &= x \\ \Phi(x, u_i^n) &= \Phi(\Phi(x, u_i), u_{i+1}^n) \end{aligned} \tag{2.1}$$

The *initial* (respectively, *current*) *state dynamical observer problem* for a finite machine $\mathcal{M}$ is to estimate $\mathcal{M}$'s *initial* (respectively, *current*) state from observations on its inputs and outputs over a finite time period. An *initial* (respectively *current*) *state dynamical observer*

takes, as input, the observed behavior of a system, i.e. a sequence of input/output pairs, and outputs an estimate of the initial (respectively *current*) state of the system.

Definition 2.2 The N-element state sequence $x_1^N \in X^N$ is an *N-consistent state sequence with respect to a given input-output sequence*, $o_1^N = [< y_1 >, < u_1, y_2 > \cdots, \langle u_{N-1}, y_N \rangle] \in Y \times (U \times Y)^{N-1}$ if the relation $CS(x_1^N)$ given by,

$$x_k = \Phi(x_1, u_1^{k-1}) \text{ and } y_k = \eta(x_k) \text{ for all } k \in [1, \cdots, N], \tag{2.2}$$

is satisfied. The set of all such sequences is denoted $CSS(o_1^N)$.

□

Definition 2.3 An *initial state estimate* set, with respect to the N-element observation sequence, o_1^N, written $ISE(o_1^N)$ or $\{\widehat{x_1}\}(o_1^N)$, is the set of initial elements of consistent state sequences corresponding to o_1^N, *i.e.*,

$$ISE(o_1^N) \equiv \{\widehat{x_1}\}(o_1^N) = \{x \in X;\ x = P_1(x_1^N) \text{ for some } x_1^N \in CSS(o_1^N)\}. \tag{2.3}$$

where $P_1(\cdot)$ denotes projection on the first component of the argument, and, analogously, a current state estimate set, with respect to the N-element observation sequence o_1^N, written $CSE(o_1^N)$ or $\{\widehat{x_N}\}(o_1^N)$, is the set of final elements of consistent states sequences corresponding to o_1^N, *i.e.*,

$$CSE(o_1^N) \equiv \{\widehat{x_N}\}(o_1^N) = \{x \in X;\ x = P_N(x_1^N) \text{ for some } x_1^N \in CSS(o_1^N)\}, \tag{2.4}$$

where $P_N(\cdot)$ denotes projection on the N-th component of the argument.

□

Definition 2.4 A finite machine $\mathcal{M} = (X, U, Y, \Phi, \eta)$ is said to be *non-anticipatively initial* (respectively *current) state observable* if there exists a sequence of non-anticipative input functions $\{u_k; u_k(o_1^k) \in U; u_k : U^{k-1} \times Y^k \to U\}$, $k = 1, 2, \cdots$, i.e., a (non-anticipative) control law u^{NA}, and a constant $K \in \mathbb{N}_+$, such that for all $x \in X$, and for all $N \geq K$ the initial (respectively, current) state estimate $\{\widehat{x_1}\}(o_1^N)$ (respectively $\{\hat{x}_N\}(o_1^N)$) is a singleton whenever o_1^N is the output resulting from the input u^{NA}.

□

In other words, there exists a past-dependent control law which forces the initial state estimate to give a single value after a finite time period.

Definition 2.5 A finite machine $\mathcal{M} = (X, U, Y, \Phi, \eta)$ is said to be *strongly initial* (respectively *current*) *state observable* if there exists a $K \in \mathbb{N}_+$, such that for all $N \geq K$, and all $u_1^N \in U^N$, the initial (respectively current) state estimate $\{\widehat{x_1}\}(o_1^N)$ (respectively $\{\hat{x}_N\}(o_1^N)$) is a singleton. The analogous *weak* notions of observability are said to hold if there exists at least one input sequence $u_1^N \in U^N$ such that the corresponding properties hold.

□

Clearly the strong i.s.o (respectively c.s.o.) property implies the non-anticipative i.s.o. (respectively c.s.o.) property holds. For any input-state-output finite machine, $\mathcal{M} =$

(X, U, Y, Φ, η) we have the property: $\mathcal{M}$ is weakly current state observable if and only if $\mathcal{M}$ is non-anticipatively current state observable (see Caines, Greiner, Wang [1991], Wang [1991]).

In the rest of this paper observability will always be taken in the strong sense.

Consider any finite machine $\mathcal{M} = (X, U, Y, \Phi, \eta)$, then for any observation sequence, $o_1^N \in O^N$, the following equations hold:

$$ISE(o_1^{N+1}) \equiv \widehat{\{x_1\}}(o_1^{N+1}) = \widehat{\{x_1\}}(o_1^N) \bigcap \Phi^{-1}(\eta^{-1}(y_{N+1}), u_1^N) \tag{2.5}$$
$$= \bigcap_{k=1}^{N} \Phi^{-1}(\eta^{-1}(y_k), u_1^{k-1}),$$

$$CSE(o_1^{N+1}) \equiv \widehat{\{x_{N+1}\}}(o_1^{N+1}) = \Phi(\widehat{\{x_N\}}(o_1^N), u_N) \bigcap \eta^{-1}(y_{N+1}) \tag{2.6}$$
$$\subseteq \bigcap_{k=1}^{N} \Phi(\eta^{-1}(y_k), u_k^{N-1}),$$

with equality in (2.6) if $\Phi(\cdot, u)$ is one to one for each $u \in U$, where in (2.5), (2.6) Φ has been extended to take sets of states in its first argument: $\Phi : \mathcal{P}(X) \times U^* \longmapsto \mathcal{P}(X)$, where $\mathcal{P}(S)$ denotes the power set of the set S by $\Phi(A, u_1^*) = \{x \in X; x = \Phi(x', u_1^*)$ for some $x' \in A\}$, Φ^{-1} denotes the inverse of Φ, $\Phi^{-1} : \mathcal{P}(X) \times U^* \longmapsto \mathcal{P}(X)$, given by $\phi^{-1}(A, u_1^*) = \{x \in X; \phi(x, u_1^*) \in A\}$ and similarly for η^{-1}, and finally $\widehat{\{x_0\}}(o_1^0)$ is defined to be X. (See Caines, Greiner, Wang [1988, 1991].) It will be noted that these equations possess the *predictor-corrector* form of many recursive algorithms in systems and control theory.

The corresponding partially ordered sets of initial and current state estimate sets will be referred to as the *initial* and *current observer trees* respectively (for the given machine). Observe that, although the state estimate sets may be identical for distinct input-output sequences, such distinct sequences uniquely define a directed acyclic graph with no confluences of edges. So, at the cost of some redundancy, we shall label the current state observation process by branches of the tree of input-output sequences, and the same is true of a COCOLOG family of theories.

Definition 2.6 A finite machine $\mathcal{M}$ is said to be *controllable to* x^T (respectively *controllable*) if for all $x \in X$ (and all $y \in Y$, respectively) there exists a sequence $u_1^{n(x,x^T)}$ (respectively $u_1^{n(x,y)}$) such that $\Phi(x, u_1^{n(x,x^T)}) = x^T$ (respectively $\Phi(x, u_1^{n(x,y)}) = y$). □

In this paper controllability is taken in the second stronger sense.

The papers Caines and Wang [1989], Caines, Greiner, Wang [1988, 1991], contain results concerning the combinatoric properties of initial and current state observer trees for any given automation. Furthermore, a dynamic programming theorem is given for a current state observable and controllable finite state machine; it states that in order to steer the system state to a target state x^T there exists a controller whose feedback law is a function only of x^T and the state estimate sets.

3 COCOLOG: Syntax and Semantics

3.1 Syntax of COCOLOG L

The COCOLOG language consists of a set of symbols $S(L)$ and specified formation rules (or syntax). The subject of the COCOLOG language is a finite machine $\mathcal{M} = (X^{\mathcal{M}}, U^{\mathcal{M}}, Y^{\mathcal{M}}, \Phi, \eta)$, where $X^{\mathcal{M}}$ is the set of states, $U^{\mathcal{M}}$ is the set of controls, $Y^{\mathcal{M}}$ is the set of outputs, Φ is a state transition function $\Phi : X^{\mathcal{M}} \times U^{\mathcal{M}} \rightarrow X^{\mathcal{M}}$ and η is a state output function, $\eta : X^{\mathcal{M}} \rightarrow Y^{\mathcal{M}}$.

We first define $S(L)$ as follows:

$$S(L) = Cons_L \bigcup Var_L \bigcup Fun_L \bigcup Apr_L \bigcup Qua_L \bigcup Lco_L \bigcup \{\perp\}.$$

The component sets of $S(L)$ are defined as follows:

Constant Symbols

$$Cons_L = \left\{x^1, \cdots, x^N\right\} \bigcup \left\{y^1, \cdots, y^p\right\} \bigcup \left\{u^1, \cdots, u^m, u^*\right\} \bigcup \left\{0, 1, \cdots, k(N), k(N)+1\right\},$$

where $k(N)+1$ will denote the upper bound on time to which our arithmetic axioms will give some of the properties of infinity. Here $k(N)$ is taken to be an arbitrary number, for example a number greater than $|X|$, or $|X|^2$, since (see Caines, Greiner, Wang [1991]) an initial, or current, state observer tree can have at most $|X|$, or $|X|^2$ respectively, non-singleton layers before it splits into nodes that will not further reduce in size.

Variable Symbols

$$Var_L = \left\{x, x', x'', \cdots, \right\} \bigcup \left\{y, y', y'', \cdots, \right\} \bigcup \left\{u, u', u'', \cdots, \right\} \bigcup \left\{l, l', \cdots, \right\}.$$

Where the variables are intended to be varying in different *sorts* or domains: the variables $x, x', x'', \cdots$ will be interpreted to represent elements in the set of states X, variables $y, y', \cdots$ will be interpreted to represent elements in the set of state output Y, and so on.

Function Symbols

$$Fun_L = \left\{\overline{\Phi}(\cdot,\cdot), \overline{\eta}(\cdot), +_L(\cdot,\cdot), -_L(\cdot,\cdot)\right\},$$

where the sort of each function symbol is defined as follows:

$\overline{\Phi}(a, b) = c$: where a and c are symbols in $\{x^1, \cdots, x^N\}$ or in $\{x, x', \cdots\}$ and b is a symbol in $\{u^1, \cdots, u^m, u^*\}$ or in $\{u, u', \cdots, \}$.

$\overline{\eta}(a) = b$: where a is a symbol in $\{x^1, \cdots, x^N\}$ or in $\{x, x', \cdots\}$ and b is a symbol in $\{y^1, y^2, \cdots, y^p\}$ or in $\{y, y', y'', \cdots\}$.

$+_L(a, b) = c$ and $-_L(a, b) = c$: where a, b and c are symbols in $\{0, 1, 2, \cdots, k(N), k(N)+1\}$ or in $\{l, l', \cdots, \}$.

Terms

The elements of the set $Term_L$ are defined via:

(i) Each constant and variable symbol is a term, i.e., $Cons_L \bigcup Var_L \subseteq Term_L$.

(ii) If t is a term and f is a function symbol, then $f(t)$ is a term.

(iii) The elements of $Term_L$ are constructed only by steps (i) and (ii) above.

Atomic Predicate Symbols $Apr_L = \left\{Eq(\cdot,\cdot), Rb\ell(\cdot,\cdot,\cdot)\right\}$.

Quantifiers $Qua_L = \left\{\forall\right\}$.

Logical Connectives $Lco_L = \left\{\rightarrow\right\}$.

Logical Constants $\{\bot\}$.

Any *well formed formula* wff of L is given by the *Backus-Naur* syntactic rule, see Goldblatt [1987]:

$$A ::= \varphi(t_1, \cdots, t_n) \mid A_1 \rightarrow A_2 \mid \bot \mid \forall v A_1;$$

where $\varphi(t_1, \cdots, t_n)\ \epsilon\ Apr_L, A_1, A_2$ are wffs and $t_1, \cdots, t_n\ \epsilon\ Term_L$, in the sense that a wff is an expression that parses according to these rules until after a finite sequence of steps one halts at elements of $S(L)$. The set of such formulas will be denoted Fma_L or L.

The other logical connectives $\neg, \vee, \wedge, \longleftrightarrow$ and the quantifier $\exists$ are defined as follows, where the parentheses (and) are used whenever they clarify the meaning of the formula:

$$\begin{aligned} \neg A &::= A \rightarrow \bot \\ A_1 \wedge A_2 &::= \neg(A_1 \rightarrow \neg A_2) \\ A_1 \longleftrightarrow A_2 &::= (A_1 \rightarrow A_2) \wedge (A_2 \rightarrow A_1) \\ A_1 \vee A_2 &::= \neg A_1 \rightarrow A_2 \\ \exists v A &::= \neg(\forall v \neg A). \end{aligned}$$

3.2 Semantics of COCOLOG L

In the following we shall distinguish symbols used in the specification of the finite machine $\mathcal{M}$ and those used in the COCOLOG language L by the convention that italic letters denote the constants and variables of $\mathcal{M}$, while COCOLOG function symbols will be denoted by a bar over the corresponding functions of $\mathcal{M}$. Following standard terminology (see e.g. Goldblatt [1987]), an *L-structure* $\mathcal{U}_L = (\mathbf{D}, I)$, or an *interpretation* I (with (*domain* $\mathbf{D}$), is a pair, where, first, $\mathbf{D} = \mathbf{X} \bigcup \mathbf{Y} \bigcup \mathbf{U} \bigcup \mathbf{I_{k(N)}}$, where $\mathbf{X}, \mathbf{Y}, \mathbf{U}$ are sets and $\mathbf{I}_{k(N)} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \cdots \mathbf{k(N)}, \mathbf{k(N)+1}\}$, is the domain of interest and, second, I is an interpretation function defined as follows:

$$I(\overline{\Phi}) = \mathbf{\Phi} : \mathbf{X} \times \mathbf{U} \rightarrow \mathbf{X},$$

$$
\begin{aligned}
I(\overline{\eta}) &= \eta : \mathbf{X} \to \mathbf{Y}, \\
I(+_L) &= +_{\mathbf{k(N)}} : \mathbf{I_{k(N)}} \times \mathbf{I_{k(N)}} \to \mathbf{I_{k(N)}}, \\
I(-_L) &= -_{\mathbf{k(N)}} : \mathbf{I_{k(N)}} \times \mathbf{I_{k(N)}} \to \mathbf{I_{k(N)}}, \\
I(c) &= \mathbf{c} \in \mathbf{D} \quad \text{where } c \in Cons_L. \\
I(\mathrm{E}q) &= \left\{ (\mathbf{t}, \mathbf{t}') \mid \mathbf{t}, \mathbf{t}' \in \mathbf{D}, \mathbf{t} = \mathbf{t}' \right\} \subseteq \mathbf{D}^2, \\
I(Rb\ell) &= \left\{ (\mathbf{x}, \mathbf{x}', \mathbf{k}) \mid \text{ there exists } \mathbf{u}_1^{\mathbf{k}} \in \mathbf{U}^k, \Phi(\mathbf{x}, \mathbf{u}_1^{\mathbf{k}}) = \mathbf{x}' \right\} \subseteq \mathbf{X}^2 \times \mathbf{I_{k(N)}}.
\end{aligned}
$$

Addition $+_{\mathbf{k(N)}}$ and subtraction $-_{\mathbf{k(N)}}$ over the finite set of integers $\{\mathbf{0}, \mathbf{1}, \mathbf{2}, \cdots \mathbf{k(N)}, \mathbf{k(N)} + \mathbf{1}\}$ are defined by the following expressions, where we follow the convention that $+_{\mathbf{k(N)}}$ and $\mathbf{k}-_{(N)}$ denote the addition and substraction in the L-structure $\mathcal{U}_L$ and $+$ and $-$ denote the standard integer arithmetical operations:

$$
\mathbf{a} +_{\mathbf{k(N)}} \mathbf{b} = \begin{cases} \mathbf{a} + \mathbf{b} & \text{if } \mathbf{a} + \mathbf{b} \leq \mathbf{k(N)} \\ \mathbf{k(N)} + \mathbf{1} & \text{if } \mathbf{a} + \mathbf{b} > \mathbf{k(N)}, \end{cases}
$$

$$
\mathbf{a} -_{\mathbf{k(N)}} \mathbf{b} = \begin{cases} \mathbf{a} - \mathbf{b} & \text{if } \mathbf{a} - \mathbf{b} \geq \mathbf{0} \\ \mathbf{k(N)} + \mathbf{1} & \text{if } \mathbf{a} - \mathbf{b} < \mathbf{0}. \end{cases}
$$

These finite integer arithmetical operations are chosen to express the dynamical properties of $\mathcal{M}$ in terms of a bounded integeral number of steps.

As usual, a *$\mathcal{U}_L$-valuation* (*I-valuation*) is a function $V : Var_L \to \mathbf{D}$ satisfying

$$
V(v) \in \begin{cases} \mathbf{X} & \text{if } v \in \{x', x'', \cdots\}, \\ \mathbf{Y} & \text{if } v \in \{y, y', y'', \cdots\}, \\ \mathbf{U} & \text{if } v \in \{u, u', u'', \cdots\}, \\ \mathbf{I_{k(N)}} & \text{if } v \in \{\ell, \ell', \ell'', \cdots\}, \end{cases}
$$

which can be extended to $V : Term_L \to \mathbf{D}$ by

$$
V(t) = \begin{cases} V(t) & \text{if } t \in Var_L, \\ I(t) & \text{if } t \in Cons_L, \\ I(f)(V(t_1), V(t_2)) & \text{if } t = f(t_1, t_2) \text{ and } f \in Fun_L. \end{cases}
$$

We take $V \sim_v V'$ to mean that V and V' are identical except in the value they assign to v and

$$
V(v/\mathbf{x}) = V' \text{ iff } V \sim_v V' \text{ and } V'(v) = \mathbf{x}
$$

$\mathcal{U}_L \models A[V]$ stands for the property that a formula A satisfies a structure $\mathcal{U}_L$ (or satisfies I) under the valuation V; this is defined recursively by:

$$
\begin{aligned}
&\mathcal{U}_L \models \mathrm{E}q(t, t')[V] \text{ iff } V(t) = V(t'), \\
&\mathcal{U}_L \models Rb\ell(x, x', k)[V] \text{ iff } (V(x), V(x'), V(k)) \in I(Rb\ell), \\
&\mathcal{U}_L \models (A_1 \to A_2)[V] \text{ iff } \mathcal{U}_L \models A_1[V] \text{ implies } \mathcal{U}_L \models A_2[V], \\
&\mathcal{U}_L \not\models \bot\,[V], \\
&\mathcal{U}_L \models \forall v A[V] \text{ iff for all } \mathbf{x} \in \mathbf{D}, \text{ it is the case that } \mathcal{U}_L \models A[V(v/\mathbf{x})].
\end{aligned}
$$

The property that a formula A is *true* in the structure $\mathcal{U}_L$, or equivalently, that I (on the domain $\mathbf{D}$) is a *model* for A is defined by

$$\mathcal{U}_L \models A \quad \text{iff} \quad \text{for all } V \text{ it is the case that } \mathcal{U}_L \models A[V];$$

conversely, A is *false*, or I (on the domain $\mathbf{D}$) is *not a model* for A, defined by:

$$\mathcal{U}_L \not\models A \quad \text{iff} \quad \text{for all } V, \text{ it is the case that } \mathcal{U}_L \not\models A[V]$$

In standard terminology, a formula A is called *valid* if it is true in all structures $\mathcal{U}_L$ (i.e. all interpretations I) if and only if for all $\mathcal{U}_L$, $\mathcal{U}_L \models A$. A formula A is *satisfiable* if there exists some structure $\mathcal{U}_L$ and some valuation V such that the satisfaction relation $\mathcal{U}_L \models A[V]$ holds. Obviously a formula A is valid if and only if $\neg A$ is unsatisfiable. Unless we relativize to a set of interpretations, the only valid formulas in a theory are those given by the logical axiom schlemata given below. This is because these must hold for any set theoretic interpretation. Other formulas, in particular the special axioms of a particular theory, may not be true under some interpretation.

3.3 Axiomatic Theory of Th_0

The formal logical theory Th_0 of the language L consists of a set of axioms, that is to say a set of formulas from Fma_L which shall be required to hold in the intended models, the set of inference rules operating on wffs of Fma_L, together with concepts of proof and theoremhood.

A general theory of finite machines is given by simply characterizing the functional property and the semi-group property of the state transition function Φ and the functional property of the output function η, namely:

(i) For any $u \in U$, any $x, x', x'' \in X^{\mathcal{M}}$ and any $y_1, y_2 \in Y^{\mathcal{M}}$, if $\Phi(x,u) = x'$ and $\Phi(x,u) = x''$, then $x' = x''$; similarly, if $\eta(x) = y_1$ and $\eta(x) = y_2$, then $y_1 = y_2$.

(ii) For all $x \in X$, all $n \in \mathbf{N}_+$ and $u_1^n \in (U^{\mathcal{M}})^n$,

$$\Phi(x, u_1^n) = \Phi(\Phi(x, u_1^{n-1}), u_n).$$

A general (partially observed) finite machine theory will be a theory true for every finite machine; such a general theory specializes whenever the transition function and the output function properties are further restricted. In this subsection we first present the axiomatic COCOLOG theory Th_0 corresponding to the information at the root node of the observer tree for a given finite machine $\mathcal{M}$. Further specializations of this theory to $Th(o_1^k)$ are obtained as observations are collected, as time proceeds, on the input-output behaviour of $\mathcal{M}$. This development is presented in the next subsection. Note that since the dynamics of $\mathcal{M}$ are known, the incomplete information aspect of $Th(o_1^k)$ is solely due to the partial observation nature of the problem.

Th_0 has a set of *logical axioms*, a set of *equality axioms* for an equality predicate, a set of *arithmetic axioms* and a set of *special axioms* which specify true facts concerning the subject the logic describes (in at least one of its interpretations). Correspondingly, $Th(o_1^k)$ is a logical

theory that has the *observation axioms*, the *state estimation axioms* and the *control axioms* (all below) added to the logical theory Th_0.

Logical Axiom Schemata

For all A, B,C ϵ Fma_L,

(i) $A \rightarrow (B \rightarrow A)$

(ii) $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$

(iii) $(\neg B \rightarrow \neg A) \rightarrow ((\neg B \rightarrow A) \rightarrow B)$ $\qquad$ $\mathbf{AXM^{log}}$

(iv) $\forall v\ A\ (v) \rightarrow A\ (t)$

(v) $\forall v\ (A \rightarrow B) \rightarrow (A \rightarrow \forall v\ B)$ $\qquad$ v not free in A.

Any formula having the same form as one of these logical axiom schemata shall be called a logical axiom. Hence the logical axiom schemata give rise to an infinite number of axioms.

Equality Axiom Schemata

In the following equality axiom schemata, P is any wff, i.e., $P \epsilon Fma_L, x, x', x'' \epsilon Var_L$ and $f \epsilon Fun_L$.

(i) $\mathrm{Eq}(x, x)$

(ii) $\mathrm{Eq}(x, x') \rightarrow \mathrm{Eq}(x', x)$

(iii) $\mathrm{Eq}(x, x'') \wedge \mathrm{Eq}(x', x'') \rightarrow \mathrm{Eq}(x, x')$ $\qquad$ $\mathbf{AXM^{eq}(L)}$

(iv) $\mathrm{Eq}(x, x') \rightarrow \mathrm{Eq}(f(x), f(x'))$ $\qquad$ for each function symbol f in L

(v) $\mathrm{Eq}(x, x') \rightarrow (P(x) \rightarrow P(x'))$ $\qquad$ for each predicate symbol P in L

Arithmetic Axiom Schemata

For any constants $\ell, \ell', \ell'' \epsilon I_{k(N)}$, if $\mathrm{l} +_{\mathbf{k(N)}} \mathrm{l}' = \mathrm{l}''$, then the following *arithmetic* axiom holds:

$$\mathrm{Eq}(\ell +_L \ell', \ell''),$$

and if $\mathrm{l} -_{\mathbf{k(N)}} \mathrm{l}' = \mathrm{l}''$, then

$$\mathrm{Eq}(\ell -_L \ell', \ell'')$$

$\mathbf{AXM^{arth}(L)}$.

□

Finite Machine Axioms

The special axioms for a given finite machine $\mathcal{M}$ are as follows: For any pair of constants $x^i, x^j \in X^{\mathcal{M}}$, and constant $u^i \in U$, if $x^j = \Phi(x^i, u^i)$ holds for $\mathcal{M}$, then the following *dynamic* axiom holds:

$$\mathrm{Eq}(\overline{\Phi}(x^i, u^i), x^j) \qquad\qquad \mathbf{AXM^{dyn}(L)}.$$

Further, for any pair of constants $x^i \in X, y^i \in Y$ such that $\eta(x^i) = y^i$ holds for $\mathcal{M}$, the following *output* axiom holds:

$$\mathrm{Eq}(\overline{\eta}(x^i), y^i) \qquad\qquad \mathbf{AXM^{out}(L)}.$$

□

The finite machine axioms given above possess an infinite number of models. It is proved below that we get a unique model (up to relabeling isomorphisms) when we impose as axioms the restrictions $|X| = N, |Y| = p$ and $|U| = m$.

Reachability Axioms

We recursively define the *reachability predicate* $Rbl(\cdot,\cdot,\cdot)$ by the following axioms:

0. $\forall x \forall x', \mathrm{Eq}(x, x') \longleftrightarrow Rbl(x, x', 0)$
1. $\forall x \forall x', (\exists u, \mathrm{Eq}(\overline{\Phi}(x, u), x')) \longleftrightarrow Rb\ell(x, x', 1)$
2. $\forall x \forall x'' \forall l, \mathrm{Eq}(\ell, k(N)) \vee \mathrm{Eq}(\ell, k(N) + 1) \vee [\{\exists x' \exists u, Rbl(x', x'', \ell) \wedge \mathrm{Eq}(\overline{\Phi}(x, u), x')\}$
   $\leftrightarrow Rbl(x, x'', \ell +_L 1)]$ $\qquad\qquad \mathbf{AXM^{Rbl}(L)}$
3. $\forall x \forall x', Rbl(x, x', k(N) + 1)$.

The reachability axioms specify the l step reachability relation $Rbl(x, x', l)$ among any pair of states x, x'. We note that in these formulas the variables x, x', x'' range over X, the variable u ranges over U and l ranges over the integers $0, 1, \cdots, k(N) + 1$. Formally Axiom 3 make all states reachable from each other in $k(N) + 1$ steps, i.e. in the number of steps that plays the rôle of infinity in our arithmetic, while Axiom 2 excludes consideration of the infinity case in order to characterize reachability on the finite numbers in the arithmetic.

Rules of Inference:

R1. Modus Ponens

$$\frac{A, A \to B}{B} \qquad ;\ where\ A, B \in Fma_L$$

R2. Generalization

$$\frac{A}{\forall v A} \qquad ;\ where\ v \in Var_L$$

□

We write Σ to denote the set of special axioms of L, i.e., $\Sigma = \{AXM^{arth}(L), AXM^{dyn}(L), AXM^{out}(L), AXM^{Rb\ell}(L)\}$. A *proof* in L is a sequence of formulas $A_1, \cdots, A_k$ in Fma_L where $A_i, 1 \leq i \leq k$, is either an axiom or a direct consequence of previous formulas via $R1$ or $R2$. The last formula A_k in the sequence is called a *theorem* and $A_1, \cdots, A_{k-1}$ is a proof of the theorem A_k. A formula A is a theorem of a first order theory with equality, written $\vdash_L A$, if, in a proof of A, only logical axioms and equality axioms have been involved; A is called a *consequence* (or *theorem*) of Σ, written $\Sigma \vdash_L A$, if, in a proof of A, axioms in Σ are involved; For brevity we write $Th_0 \equiv Th_0(L)$ for the set of theorems of Σ; hence we have $Th_0 = \{A : \Sigma \vdash_L A\}$ and we use the standard notation $Th_0 \vdash A$, which is customarily read as A is a *theorem* of, or is *provable* (*derivable*) in, the theory Th_0.

A structure $\mathcal{U}_L$ (i.e. an interpretation I on a domain **D**) of the theory Th_0 is called a *model* of the theory if and only if all the axioms of Th_0 are interpreted *true* in $\mathcal{U}_L$, i.e. if and only if I is a model for each axiom of Th_0.

Example 3.1 The following is a simple example illustrating the notion of the logical control theory Th_o in COCOLOG. The finite machine $\mathcal{M} = (X^{\mathcal{M}}, U, Y, \Phi, \eta)$ is given in Figure 1, where $X^{\mathcal{M}} = \{x^1, x^2, x^3\}, U = \{u^1, u^2\}, Y = \{y^1, y^2\}$, $\eta(x^1) = \eta(x^2) = y^1, \eta(x^3) = y^2$ and Φ is given explicitly in the graph in the figure.

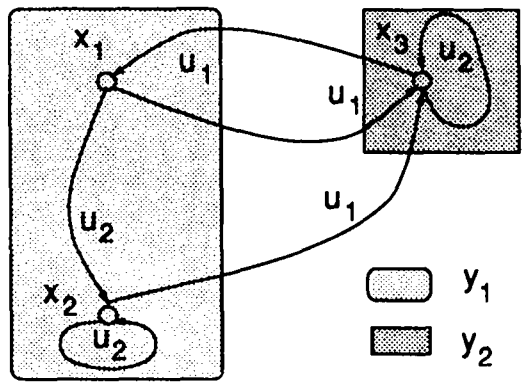


Figure 1: A Three State Machine

The COCOLOG system for this finite machine $\mathcal{M}$ consists of a tree of first order theories $Th_0, Th(o_1^1), Th(o_1^2), \cdots$. Here we deal only with the theory Th_0. We shall take $k(N) = k(3) = |X^{\mathcal{M}}|^2 = 9$ and so "infinity" for the finite arithmetic of this theory is $k(N) + 1 = 10$.

The general logical axioms, the equality axioms, the axioms of reachability and the rules of inference are as given above and the special axioms of the machine are given explicitly as follows:

$$\mathrm{Eq}(\overline{\Phi}(x^3, u^1), x^1)\ \mathrm{Eq}(\overline{\Phi}(x^1, u^2), x^2)\ \mathrm{Eq}(\overline{\Phi}(x^2, u^2), x^2)$$
$$\mathbf{AXM^{dyn}(L_3)}$$
$$\mathrm{Eq}(\overline{\Phi}(x^1, u^1), x^3)\ \mathrm{Eq}(\overline{\Phi}(x^2, u^1), x^3)\ \mathrm{Eq}(\overline{\Phi}(X^3, u^2), x^3)$$
$$\mathrm{Eq}(\overline{\eta}(x^1), y^1)\ \mathrm{Eq}(\overline{\eta}(x^2), y^1)\ \mathrm{Eq}(\overline{\eta}(x^3), y^2) \qquad \mathbf{AXM^{out}(L_3)}.$$

The set of theorems of Th_0 is exactly the set of true formulas of L_3, as guaranteed by the general completeness result proved below.

To illustrate logical deduction in COCOLOG we shall give a proof of the theorem $Rb\ell(x^1, x^3, 2)$ in theory Th_0; this theorem asserts that the state $\mathbf{x^1}$ is controllable to the state $\mathbf{x^3}$ in two steps. Note that we can verify from the model in Figure 1, that $Rb\ell(x^1, x^3, 2)$, is true in Th_0.

Proof of $Rbl(x^1, x^3, 2)$

1.	$\mathrm{E}q(\overline{\Phi}(x^2, u^1), x^3)$	Finite Machine Axiom
2.	$\exists u, \mathrm{E}q(\overline{\Phi}(x^2, u), x^3)$	1 and $AXM^{log}(iv)$
3.	$Rb\ell(x^2, x^3, 1)$	2 and $AXM^{Rb\ell}(L)$ (1) and MP
4.	$\mathrm{E}q(\overline{\Phi}(x^1, u^2), x^2)$	Finite Machine Axiom
5.	$\exists u, \mathrm{E}q(\overline{\Phi}(x^1, u), x^2)$	4 and $AXM^{log}(iv)$
6.	$\exists u, Rb\ell(x^2, x^3, 1) \wedge \mathrm{E}q(\overline{\Phi}(x^1, u), x^2)$	3 and 5
7.	$\mathrm{E}q(2, 1 +_L 1)$	Arithmetic Axiom
8.	$Rb\ell(x^1, x^3, 2)$	$6, 7, AXM^{Rb\ell}(L_3)(2)$ and MP

□

3.4 The COCOLOG Language $L(o_1^k)$: Syntax and Semantics

The language $L^k \triangleq L(o_1^k)$ is an extension of the language L obtained by adding new function symbols and atomic predicates in the following way:

$$S(L^k) \triangleq L(o_1^k) = L \bigcup_{j=1}^{k} Cons_{Lj} \bigcup_{j=1}^{k} Apr_j, \qquad Fma_{L^k}$$

where $Apr_j = \left\{CSE_j(\cdot)\right\}$ and $Cons_{Lj} = \{U(j), Y(j)\}$. The sort of each new function symbol in L^k is given by:

$U(j)$ is a symbol in $\{u^1, \ldots, u^m\}$

$Y(j)$ is a symbol in $\{y^1, \cdots, y^p\}$.

Set $Fma_{L^0} = Fma_L$; then the set of well formed formulas Fma_{Lj} for $j \geq 1$ is defined by:

$$A ::= CSE_k(x) \mid B \mid A' \rightarrow A'' \mid \forall v A',$$

where $B \in Fma_{Lj-1}$, and $A', A'' \in Fma_{Lj}$.

As before, an L_k-structure $\mathcal{U}_{L^k} = (I_k, \mathbf{D})$ is a pair, where the interpretation function I_k is an extension of I by:

$$I_k(CSE_k) = \left\{\mathbf{x}; \mathbf{x} \in \widehat{\{x_k\}}(o_1^k) \subseteq \mathbf{X}\right\}$$
$$I_k(U(k)) = \mathbf{U(k)} \in \mathbf{D} \text{ where } \mathbf{U}(k) \in Cons_{L^k},$$
$$I_k(Y(k)) = \mathbf{Y(k)} \in \mathbf{D} \text{ where } Y(k) \in Cons_{L^k}.$$

The satisfaction relation $\mathcal{U}_{L^k} \models A[V]$ is an extension of $\mathcal{U}_{L^{k-1}} \models A[V]$ obtained by adding the following definitions:

$$\begin{aligned}
&\mathcal{U}_{L^k} \models CSE_k(x)[V] \text{ iff } V(x) \in \widehat{\{x_k\}}(o_1^k) \text{ where } o(1) = y(1),\\
&\qquad o(2) = (u(1), y(2)), \cdots, o(k) = (u(k-1), y(k)),\\
&\mathcal{U}_{L^k} \models B[V] \text{ iff } \mathcal{U}_{L^k} \models B[V] \text{ for any } B \in Fma_{L^{k-1}},\\
&\mathcal{U}_{L^k} \models \forall v A[V] \text{ iff for all } \mathbf{x} \in \mathbf{D}, \text{ it is the case that}\\
&\qquad \mathcal{U}_L \models A[V](v/\mathbf{x}) \text{ for any } A \in Fma_{L^k},\\
&\mathcal{U}_{L^k} \models (A_1 \to A_2)[V] \text{ iff } \mathcal{U}_{L^k} \models A_1[V] \text{ implies } \mathcal{U}_{L^k} \models A_2[V].
\end{aligned}$$

The properties *true* and *false* for a formula and the concept of a *model* for a theory $Th(o_1^k)$ are defined in analogy with those of Th_0.

3.5 Axiomatic Theory of $Th(o_1^k)$

At each instant k the observer receives $u(k-1)$ and $y(k)$; for the constants u^i and y^i such that $u^i = u(k-1)$ and $y^i = y(k)$, the following formulas (i) and (iii) express the fact that those observations are added incrementally as axioms to Th_0 to form the theory $Th(o_1^k)$ of L^k.

Observation Axioms

For $k \geq 1$, and subject to the convention that axiom (ii) below holds only in case $k > 1$,

(i) $\mathrm{Eq}(Y(k), y^i)$

(ii) $\mathrm{Eq}(U(k-1), u^i)$ $\qquad\qquad$ $\mathbf{AXM^{obs}(L^k)}$,

□

It will turn out that in the formulation given in this paper the second axiom $\mathbf{AXM^{obs}(L^k)}$ is redundant; this is because the control axioms $\mathbf{AXM^{cntl}(L^{k-1})}$ assign the value of the constant $U(k-1)$ at $k-1$ in the theory $Th(o_1^{k-1})$ and by the definition of $S(L^k), S(L^{k+1}), \cdots$ this value is inherited by all subsequent theories.

State Estimation Axioms

The following are the set of *Axioms of Conditional State Estimation.* They express in axiomatic form the recursive formulas (2.6) for the current state estimate sets.

In case $k = 1$:

$$\begin{aligned}
&\mathrm{Eq}(\overline{\eta}(x^1), Y(k)) \longleftrightarrow CSE_1(x^1)\\
&\qquad\vdots \qquad\quad \vdots \qquad\quad \vdots \qquad \mathbf{AXM^{est}(L^k)}\\
&\mathrm{Eq}(\overline{\eta}(x^N), Y(k)) \longleftrightarrow CSE_1(x^N).
\end{aligned}$$

□

In case $k > 1$:

$$\begin{aligned}
&\exists x, CSE_{k-1}(x) \bigwedge \mathrm{Eq}(\overline{\Phi}(x, U(k -_L 1)), x^1) \bigwedge \mathrm{Eq}(\overline{\eta}(x^1), Y(k)) \\
&\qquad\qquad \longleftrightarrow CSE_k(x^1) \\
&\qquad\qquad \vdots \qquad\qquad\qquad\qquad\qquad\qquad \mathbf{AXM^{est}(L^k)} \\
&\exists x, CSE_{k-1}(x) \bigwedge \mathrm{Eq}(\overline{\Phi}(x, U(k -_L 1)), x^N) \bigwedge \mathrm{Eq}(\overline{\eta}(x^N), Y(k)) \\
&\qquad\qquad \longleftrightarrow CSE_k(x^N).
\end{aligned}$$

□

Control Axioms

The following is the general form of a set of *Conditional Control Axioms*, where $C_j(\cdot)$ is a *conditional formula* expressible in terms of $Fma_{L(o_1^k)}$:

$$\begin{aligned}
C_1(Fma_{L^k}) &\longrightarrow \mathrm{Eq}(U(k), u^1) \\
\neg C_1(Fma_{L^k}) \bigwedge C_2(Fma_{L^k}) &\longrightarrow \mathrm{Eq}(U(k), u^2) \\
\vdots \qquad\qquad &\longrightarrow \qquad \vdots \qquad \mathbf{AXM^{cntl}(L^k)} \\
\bigwedge_{j=1}^{m-1} (\neg C_j(Fma_{L^k})) \bigwedge C_m(Fma_{L^k}) &\longrightarrow \mathrm{Eq}(U(k), u^m) \\
\bigwedge_{j=1}^{m} (\neg C_j(Fma_{L^k})) &\longrightarrow \mathrm{Eq}(U(k), u^*).
\end{aligned}$$

□

This set of axioms is central to the construction of COCOLOG. They have the following interpretation: If the condition $C_1(Fma_{L^k})$ is provable in the theory $Th(o_1^k)$, then invoking the first axiom, we obtain the defined constant value u^1 as the value of the control constant $U(k)$; if not, but if $C_2(Fma_{L^k})$ can be proved, then the second axiom gives the defined value u^2 to the control constant $U(k)$; and so on. If none of the conditions $C_1, C_2, \cdots, C_m$ hold, then the last axiom sets the control function equal to the arbitrary constant u^*. This procedure uniquely determines the value of $U(k)$. When $k \rightarrow k+1$, we make the meta-logical step of passing to the theory $Th(o_1^{k+1})$ carrying along all the previous axioms including the constant value u^i chosen above (see Figure 2). This is formally enforced by the following definition of the axiom set generating $Th(o_1^k)$:

$$\Sigma^k = \Sigma \bigcup_{j=1}^{k} \{\mathbf{AXM^{obs}(L^j)}, \mathbf{AXM^{est}(L^j)}, \mathbf{AXM^{cntl}(L^j)}\},$$

where Σ is the axiom generating Th_0. Hence, in the new $Th(o_1^{k+1})$, the observed control action $U(k)$ is precisely the constant value u^i determined in $Th(o_1^k)$.

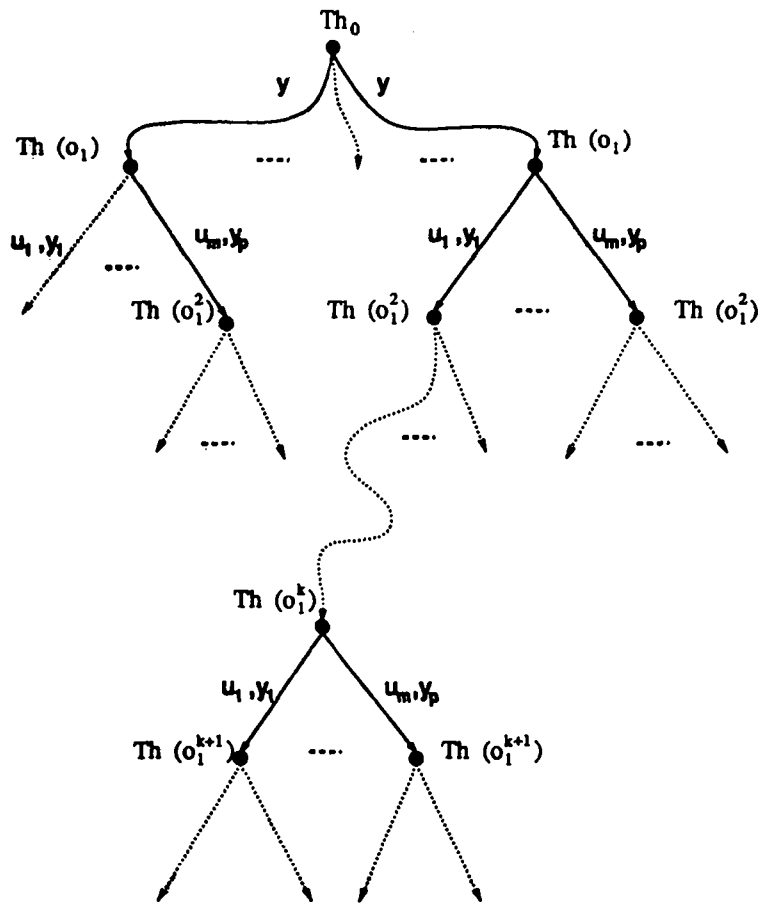


Figure 2: A COCOLOG Tree of Logical Theories

Example 3.2 A set of Conditional Control Axioms which display a specific control law, is the following.

Consider the machine $\mathcal{M} = (X^{\mathcal{M}}, U^{\mathcal{M}}, Y^{\mathcal{M}}, \Phi, \eta), Y^{\mathcal{M}} = X^{\mathcal{M}}, \eta = id$, together with the control objective: steer the current state of the system to x^T. For this example, the State Estimation Axioms, $\mathbf{AXM^{est}(L^k)}$ yield:

$\mathbf{k = 1}$ Let the initial state of the system $\mathcal{M}$ be x^j, then $Y(1) = y^j = x^j$. In this case $AXM^{est}(L^1)$ gives $CSE(x^j)$ and $1 \leq p \leq N$.

$\mathbf{k = 2}$ Let $U(1) = u^s$ and $\overline{\Phi}(x^j, u^s) = x^\ell$. Then $Y(2) = y^\ell \equiv x^\ell$. So taking $u = u^s, y = y^\ell, x = x^j$ we have

$$CSE_2(x^j) \wedge \mathrm{Eq}(\overline{\Phi}(x^j, u^s), x^\ell) \wedge \mathrm{Eq}(u(1), u^s) \wedge \mathrm{Eq}(\overline{\eta}(x^\ell), y^\ell) \wedge \mathrm{Eq}(Y(2), y^\ell)$$

and hence

$$\exists u \exists y \exists x, CSE_1(x) \wedge \mathrm{Eq}(\overline{\Phi}(x, u), x^\ell) \wedge \mathrm{Eq}(U(1), u) \wedge \mathrm{Eq}(\overline{\eta}(x^\ell), y) \wedge \mathrm{Eq}(Y(2), y)$$

is satisfied and so we have

$$CSE_2(x^\ell) \text{ and } \neg CSE_2(x^m), m \neq \ell, 1 \leq m \leq N.$$

Let us adopt the following abbreviation: for $CSE_k(x(k)) \wedge \mathrm{E}q(\Phi(x(k),u),x')$, where $x(k)$ is a state name, we shall write $\mathrm{E}q(\Phi(x(k),u),x')$. Then set

$$\begin{aligned} C_1^* &= \exists x' \exists \ell, \mathrm{E}q(\Phi(x(k),u^1),x') \wedge Rb\ell(x',x^T,\ell) \\ C_2^* &= \exists x' \exists \ell, \mathrm{E}q(\Phi(x(k),u^2),x') \wedge Rb\ell(x',x^T,\ell) \\ &\vdots \\ C_m^* &= \exists x' \exists \ell, \mathrm{E}q(\Phi(x(k),u^m),x') \wedge Rb\ell(x',x^T,\ell) \end{aligned}$$

and

$$C_0^* = \neg Rb\ell(x(k),x^T,0) \equiv \neg \mathrm{E}q(x(k),x^T).$$

Then set

$$\begin{aligned} C_1 &= C_0^* \wedge C_1^* \\ &\vdots \qquad \vdots \\ C_m &= C_0^* \wedge C_m^* \end{aligned}$$

and arrange these conditional formulas into the axiom schema $AXM^{cntl}(L^k)$.

We shall assume that $\mathcal{M}$ is such that in any state $x^j \in X$, an extra $m+1$-th control u^* has the effect of keeping the system at x^j, i.e. $\Phi(x^j,u^*) = x^j, 1 \leq j \leq |X|$.

The j-th conditional formula C_j states that there exists a path of length greater than or equal to one from the current state $x(k)$ to the target state x^T, and a control u^j either steers the system to x^T in one step or is an initial control of a sequence (of length greater than one) that steers the system state to x^T. (Notice that the case of a path of length one corresponds to $\ell = 0$.) The corresponding Conditional Control Axiom states that C_k holds for no control index k strictly less than j but C_j itself holds. We observe that if the control u^* is used at the instant k by this COCOLOG controller it indicates that either $x(k) = x^T$ or x^T is not reachable from $x(k)$.

Further inspection of the conditional formulas and the CCAs shows that they allow the possibility, for certain machines, that the system would maintain itself in a sequence of states from which x^T was reachable in a fixed number of steps $L(L > 1)$ without ever actually converging to x^T. Such apparently perverse behaviour may be prevented by the following elaboration of C_j^* for $1 \leq j \leq m$:

$$\begin{aligned} C_j^* = \mathrm{E}q(\Phi(x(k),u^j),x^T) \bigvee \\ \big[\exists x' \exists \ell \forall s \big\{ \neg \mathrm{E}q(\ell,0) \wedge \big[\neg \mathrm{E}q(\ell - s, k(N)+1) \bigvee \neg Rb\ell(x(k),x^T,s)\big] \\ \wedge \mathrm{E}q(\Phi(x(k),u^j),x') \wedge Rb\ell(x',x^T,\ell) \ \big\}\big], \end{aligned}$$

with the final version of the conditional formulas being $C_j = C_0^* \wedge C_j^*, 1 \leq j \leq m$. (The literal $\neg \mathrm{E}q(\ell,0)$ is only included for clarity, as the case $\ell = 0$ is excluded by the formula C_0^*.)

4 Consistency, Completeness and Decidability of COCOLOG Theories

We say a set of formulas T is consistent with respect to the *first order theory with equality* if there does not exist a formula A where A and $\neg A$ are both derivable from T. A first order theory with equality generated by a set of special axioms Σ is *complete* if any formula true in every model of Σ is deducible from Σ.

The consistency and completeness of the axiomatic theory presented at Section 3 can be established by a simple application of classical results on the consistency and completeness of first order theories with equality.

We consider the following generalized form of completeness theorem for the theory Th_0. As before, we denote by Σ the set $\{AXM^{dyn}(L), AXM^{obs}(L), AXM^{rbl}(L)\}$ of special axioms of Th_0; if $\mathcal{A}$ is a consequence of Σ under a first order theory with equality then this is written as $\Sigma \vdash_L \mathcal{A}$, and $\mathcal{M}_\Sigma$ will denote any model for Σ.

Theorem 4.1 (Soundness) *For any formula $A \in Fma_L$ and any model $\mathcal{M}$ of Σ we have:*

$$\Sigma \vdash_L A \text{ implies } \mathcal{M}_\Sigma \models A.$$

Proof

Soundness follows from the fact that the axioms Σ are true formulas in every model $\mathcal{M}$ of Σ and the rules of inference preserve truthfulness, hence all deducible theorems from the axioms will be true in any model of the axioms. □

A set of formulas T is *absolutely consistent* with respect to a first order theory with equality if and only if there exists some formula which is not derivable from T, i.e., $\exists A, T \nvdash_L A$.

Theorem 4.2 (Equivalence) *For any first order theory with equality where Modus Ponens is a rule of inference, the following are equivalent:*

(i) *T is consistent*, i.e. $T \nvdash_L \bot$

(ii) *T is absolutely consistent*, i.e. $\exists A, T \nvdash_L A$.

Proof

(i) $\Longrightarrow$ (ii): (i) is equivalent to the statement that for any $A, T \vdash_L A$ semantically implies $T \nvdash_L \neg A$ (since otherwise one has $T \vdash_L A$ and $T \vdash_L \neg A$ which is $A \rightarrow \bot$ and so $T \vdash_L \bot$ by Modus Ponens). This implies there exists a formula A such that $T \nvdash_L A$ and hence (i) implies (ii).

(ii)$\Longrightarrow$ (i): (i) is false if and only if for some formula A, $T \vdash_L A$ and $T \vdash_L \neg A$. Now take any formula $B \in Fma_L$. Then we have $\vdash_L \neg A \rightarrow (\neg B \rightarrow \neg A)$ and $T \vdash_L \neg A$. Hence, by Modus Ponens, we have $T \vdash_L \neg B \rightarrow \neg A$.

In the same way, we have $\vdash_L A \rightarrow (\neg B \rightarrow A)$ and $T \vdash_L A$, hence we have $T \vdash_L \neg B \rightarrow A$. Again by Modus Ponens and the third logical axiom we get $T \vdash_L B$ and hence we get the negation of (ii) as required.

□

Theorem 4.3 (Consistency) *Σ is consistent with respect to the first order theory with equality, i.e., $\Sigma \nvdash_L \perp$.*

Proof

This follows from the existence of the model $\mathcal{M}_\Sigma$ for the set of axioms Σ. Take any formula $A \in \Sigma$ which is an axiom of Th_0. Then we have $\mathcal{M}_\Sigma \not\models \neg A$. By the soundness theorem, this implies $\Sigma \nvdash_L \neg A$. By the Theorem 4.4, this implies $\Sigma \nvdash_L \perp$ and hence Σ is consistent.

□

Theorem 4.6 Generalized Completeness *A formula $A \in Fma_L$ is true in every model $\mathcal{M}_\Sigma$ of Σ if and only if A is a consequence of Σ under the first order theory with equality, i.e.*

$$\mathcal{M}_\Sigma \models A \quad \textit{iff} \quad \Sigma \vdash_{L,\Sigma} A.$$

Proof

We only prove the *completeness* part of the theory here, the *soundness* part follows from the Theorem 4.1

Suppose $\Sigma \nvdash_L A$, we need to show there exists a model $\mathcal{M}_\Sigma$ such that $\mathcal{M}_\Sigma \not\models A$.

$\Sigma \cup \{\neg A\}$ is consistent since Σ is consistent and the assumption is that $\Sigma \nvdash_L A$. Now by Henkin's Theorem (see e.g. Mendelsen [1964]) states that every consistent set of first order formulas has a model. Hence there exists a model $\mathcal{M}^*_{\Sigma \cup \{\neg A\}}$ for $\Sigma \cup \{\neg A\}$. But this model is also a model $\mathcal{M}_\Sigma \triangleq \mathcal{M}^*_{\Sigma \cup \{\neg A\}}$for Σ and clearly $\mathcal{M}_\Sigma \not\models A$, as required for Σ.

□

Next, we construct the unique model property for Σ and therefore we get decidability of the theoremhood for COCOLOG theorems.

As we mentioned before, we can get a unique model by adding additional axioms to specify sizes of X, U, Y. Otherwise, there can be infinitely many different models. For example, any finite or infinite machine $\mathcal{M}' = (X, U, Y, \Phi', \eta')$ satisfying $X \subseteq X', U \subseteq U', Y \subseteq Y'$ and such that Φ' and η' are compatible with Φ and η on X, Y and U, can be a model of the given machine axioms. Hence the machine axioms alone cannot uniquely characterize a given finite machine. In fact, one cannot determine a unique model by any given set of axioms. The most one can achieve by axiomatization is a set of equivalent models up to isomorphism. Hence uniqueness is only used in this sense.

Suppose $|X^{\mathcal{M}}| = N$, $|U^{\mathcal{M}}| = m$ and $|Y^{\mathcal{M}}| = p$, we first consider the *size axioms* for $X^{\mathcal{M}}$, then we can derive the size axioms for $U^{\mathcal{M}}$ and $Y^{\mathcal{M}}$ respectively in a similar manner.

The Size Axiom of $X^{\mathcal{M}}$

$$\begin{aligned} X_N^{\mathcal{M}} :&\neg Eq(x^1,x^2) \wedge \neg Eq(x^1,x^3) \wedge Eq(x^1,x^4) \wedge \cdots \wedge \neg Eq(x^1,x^N) \\ &\wedge \neg Eq(x^2,x^3) \wedge \neg Eq(x^2,x^4) \wedge \cdots \wedge \neg Eq(x^2,x^N) \\ &\wedge \neg Eq(x^3,x^4) \wedge \cdots \wedge \neg Eq(x^3,x^N) \\ &\vdots \\ &\wedge \neg Eq(x^{N-1},x^N) \end{aligned}$$

$X_N^{\mathcal{M}}$ specify the fact that there are at least N distinct constant symbols in the state space $X^{\mathcal{M}}$ of the finite machine $\mathcal{M}$, i.e., $|X^{\mathcal{M}}| \geq N$.

Next we specify the fact there are at most N elements in the intended model by $\neg X_{N+1}^{\mathcal{M}}$.

$$\neg X_{N+1}^{\mathcal{M}} :\forall x\Big(\bigvee_{i=1}^{N} Eq(x,x^i)\Big).$$

By adding $X_N^{\mathcal{M}}$ and $\neg X_{N+1}^{\mathcal{M}}$ to the originally proposed machine axioms the only models one can get will be the models that have exactly N distinct states. That is the set of N-state machines in which Φ and η are given as specified. Further, if we add restrictions on the size of $U^{\mathcal{M}}$ and $Y^{\mathcal{M}}$ we get a unique model for $\mathcal{M}$.

In the following we denote $\mathcal{M}$ and $\mathcal{M}'$ as finite machines and also let $\mathcal{M}, \mathcal{M}'$ denote the sets $\mathcal{M} = X^{\mathcal{M}} \bigcup U^{\mathcal{M}} \bigcup Y^{\mathcal{M}}$ and $\mathcal{M}' = X^{\mathcal{M}'} \bigcup U^{\mathcal{M}'} \bigcup Y^{\mathcal{M}'}$.

Definition 4.1 If $\mathcal{M}$ and $\mathcal{M}'$ are two finite machines, then a map h from $\mathcal{M}$ to $\mathcal{M}'$ is called a homomorphism *if*

$$\begin{aligned} h(\Phi(\mathbf{x},\mathbf{u})) &= \Phi'(h(\mathbf{x}),h(\mathbf{u})), \\ h(\eta(\mathbf{x})) &= \eta'(h(\mathbf{x})), \\ h(+_{k(\mathbf{N})}(\mathbf{l},\mathbf{l}')) &= +'_{k(\mathbf{N})}(h(\mathbf{l}),h(\mathbf{l}')), \\ h(-_{k(\mathbf{N})}(\mathbf{l},\mathbf{l}')) &= -'_{k(\mathbf{N})}(h(\mathbf{l}),h(\mathbf{l}')). \end{aligned}$$

h is called an *isomorphism* if there exists a homorphism h' from $\mathcal{M}'$ to $\mathcal{M}$ such that the composition $h' \circ h$ of h' and h is the identity on $\mathcal{M}$. □

If two L-structures $\mathcal{U}_L, \mathcal{U}'_L$ for Th_0 are such that the machines $\mathcal{M}, \mathcal{M}'$ are isomorphic we say $\mathcal{U}_L, \mathcal{U}'_L$, have *isomorphic pre-interpretations.* If the relations on the domains corresponding to the predicates $Eq(\cdot,\cdot)$ and $Rb\ell(\cdot,\cdot,\cdot)$ are also isomorphic, we say the *interpretations* or *models*, are *isomorphic.*

Define

$$\Sigma_{\mathcal{M}}^0 = \Sigma \bigcup X_N^{\mathcal{M}} \bigcup \neg X_{N+1}^{\mathcal{M}} \bigcup U_m^{\mathcal{M}} \bigcup \neg U_{m+1}^{\mathcal{M}} \bigcup Y_p^{\mathcal{M}} \bigcup \neg Y_{p+1}^{\mathcal{M}}$$

as the set of axioms for the given finite machine $\mathcal{M}$, at the instant zero.

Theorem 4.5 (Unique Model Property) *The logical theory generated by $\Sigma^0_{\mathcal{M}}$ has a unique model up to isomorphism.*

Proof

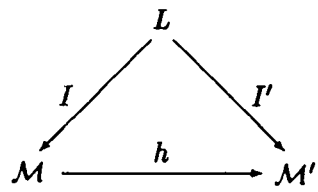
First we establish that all pre-interpretations are isomorphic by showing the existence of a homomorphic mapping between any given pair of models of $\Sigma^0_{\mathcal{M}}$.

Now consider any two models $\mathcal{M}$ and $\mathcal{M}'$ where $\mathcal{M} = (X, Y, U, \Phi, \eta)$ and $\mathcal{M}' = (X', Y', U', \Phi', \eta')$. By the size axioms we have $|X| = |X'| = N$, $|Y| = |Y'| = p$ and $|U| = |U| = m$. Then by the machine axioms we have $\Phi : X \times U \to X$ and $\Phi' : X \times U \to X$; $\eta : X \to Y$ and $\eta' : X \times U \to X$. Now an one-to-one and onto mapping $h : \mathcal{M} \to \mathcal{M}'$ can be defined, where $\mathcal{M}$ here is also taken as the union of X, U and Y and $\mathcal{M}'$ is also taken as the union of X, U and Y.

Let L denote the set of symbols of logical theory for a finite machine, $I : L \to \mathcal{M}$ and $I' : L \to \mathcal{M}'$ be the interpretation functions correspond to the model $\mathcal{M}$ and $\mathcal{M}'$ respectively. Construct a mapping $h : \mathcal{M} \to \mathcal{M}'$ such that the following relation is satisfied:

$$h(m) = I'(I^{-1}(m)) \qquad \text{for any } m \in \mathcal{M}$$

The relations among the set of $L, \mathcal{M}, \mathcal{M}'$ and the mappings of I, I' and h are shown as follows:



We need to show that h is a bijective mapping. This property is guaranteed by the bijective property of the mappings I and I'.

First, the onto property can be shown by taking any $m' \in \mathcal{M}'$, then we have $I'^{-1}(m') = l$ for some $l \in L$ and $I(l) = m$ for some $m \in \mathcal{M}$. This m is the preimage of the m' under h because

$$h(m) = I'(I^{-1}(m)) = I'(\ell) = I'(I'^{-1}(m')) = m'.$$

Similarly, the one-to-one property is immediately obtained from

$$h(m_1) = h(m_2) \text{ iff } I'(I^{-1}(m_1)) = I'(I^{-1}(m_2)) \text{ iff } m_1 = m_2.$$

Now let $h(m) = m'$ for some $m \in \mathcal{M}$ and $m' \in \mathcal{M}'$ and take some dynamical formula $Eq(\overline{\Phi}(x^i, u^i), x^j)$ from the language L. The interpretation I will map this formula to $\Phi(I(x^i), I(u^i)) = I(x^j)$ which is $\Phi(\mathbf{x}^{\mathbf{i}}_{\mathbf{m}}, \mathbf{u}^{\mathbf{i}}_{\mathbf{m}}) = \mathbf{x}^{\mathbf{j}}_{\mathbf{m}}$ and the interpretation I' will map the formula to $\Phi'(I'(x^i), I'(u^i)) = I'(x^j)$ which is $\Phi'(\mathbf{x}^{\mathbf{i}}_{\mathbf{m}'}, \mathbf{u}^{\mathbf{i}}_{\mathbf{m}'}) = \mathbf{x}^{\mathbf{j}}_{\mathbf{m}'}$. Then since $h(m) = I'(I^{-1}(m))$ we have the following relationship between the two models:

$$\Phi(\mathbf{x}^{\mathbf{i}}_{\mathbf{m}}, \mathbf{u}^{\mathbf{i}}_{\mathbf{m}}) = \mathbf{x}^{\mathbf{j}}_{\mathbf{m}} \text{ iff } \Phi'(\mathbf{x}^{\mathbf{i}}_{\mathbf{m}'}, \mathbf{u}^{\mathbf{i}}_{\mathbf{m}'}) = \mathbf{x}^{\mathbf{j}}_{\mathbf{m}'}.$$

Similarly, reference to the observation axioms $AXM^{out}(L)$ yields

$$\eta(\mathbf{x}^{\mathbf{i}}_{\mathbf{m}}) = \mathbf{y}^{\mathbf{i}}_{\mathbf{m}} \text{ iff } \eta'(\mathbf{x}^{\mathbf{i}}_{\mathbf{m}'}) = \mathbf{y}^{\mathbf{i}}_{\mathbf{m}'}.$$

For the finite arithmetic of Th_0 things are yet simpler since the properties of $+_L, -_L$ were defined in terms of $+_{\mathbf{k(N)}}, -_{\mathbf{k(N)}}$ on the integers. Hence trivially

$$+_{k(N)}(\mathbf{l}_{\mathbf{m}}, \mathbf{l}'_{\mathbf{m}}) = \mathbf{l}''_{\mathbf{m}} \text{ iff } +'_{k(N)}(\mathbf{l}_{\mathbf{m}'}, \mathbf{l}'_{\mathbf{m}'}) = \mathbf{l}''_{\mathbf{m}'}$$
$$-_{k(N)}(\mathbf{l}_{\mathbf{m}}, \mathbf{l}'_{\mathbf{m}}) = \mathbf{l}''_{\mathbf{m}} \text{ iff } -'_{k(N)}(\mathbf{l}_{\mathbf{m}'}, \mathbf{l}'_{\mathbf{m}'}) = \mathbf{l}''_{\mathbf{m}'}$$

From this it follows immediately that the bijection, h yields the desired isomorphism i.e., h is a bijection satisfying

$$h(\Phi(\mathbf{x}^{\mathbf{i}}, \mathbf{u}^{\mathbf{i}})) = \Phi'(h(\mathbf{x}^{\mathbf{i}}), h(\mathbf{u}^{\mathbf{i}})),$$
$$h(\eta(\mathbf{x}^{\mathbf{i}})) = \eta'(h(\mathbf{x}^{\mathbf{i}})),$$
$$h(+_{k(N)}(\mathbf{l}, \mathbf{l}')) = +'_{k(N)}(h(\mathbf{l}), h(\mathbf{l}')),$$
$$h(-_{k(N)}(\mathbf{l}, \mathbf{l}')) = -'_{k(N)}(h(\mathbf{l}), h(\mathbf{l}')).$$

It follows that $\Sigma^0_{\mathcal{M}}$ has a unique pre-interpretation up the isomorphism. Evidently, the equality predicate has a unique interpretation on the domain of the pure-interpretation. Finally, the recursive nature of the reachability axioms $AXM^{rbl}(L)$ reveals that there is a unique relation on the elements of the (proven unique) pre-interpretation that satisfies the reachability axioms.

Definition 4.2 (Proper Formula) A formula P is a proper formula with respect to a set of formula Γ if P contain neither any predicate symbols nor function symbols which do not appear in any formulas in Γ □

Definition 4.3 (Complete Axiomatization) A set of formulas Γ is said to be complete if either P or $\neg P$ is a consequence of Γ for any proper formula P with respect to Γ □

It is known result that if a set of axioms has a unique model then that set of axioms is complete. We state this in the following theorem.

Theorem 4.6 The axiomatization defined by $\Sigma^0_{\mathcal{M}}$ is a complete axiomatization of $\mathcal{M}$.
Proof

To prove that $\Sigma^0_{\mathcal{M}}$ is a complete axiomatization of $\mathcal{M}$, we need to show that for any formula $A \in L$ either $\Sigma^0_{\mathcal{M}} \vdash A$ or $\Sigma^0_{\mathcal{M}} \vdash \neg A$ is true. We know $\Sigma^0_{\mathcal{M}}$ is consitent by the existence of a model for $\Sigma^0_{\mathcal{M}}$. By Lindenbaum's lemma, see Mendelson [1964], if $\Sigma^0_{\mathcal{M}}$ is a consistent first order theory, then there is a consistent complete extension of $\Sigma^0_{\mathcal{M}}$. But since we know $\Sigma^0_{\mathcal{M}}$ has a unique model, the complete extension of $\Sigma^0_{\mathcal{M}}$ is $\Sigma^0_{\mathcal{M}}$. Hence $\Sigma^0_{\mathcal{M}}$ is complete, since otherwise $\Sigma^0_{\mathcal{M}}$ could not have a unique model. □

Theorem 4.7 (Decidable Theoremhood) *The logical theory as generated by $\Sigma^0_{\mathcal{M}}$ for any given finite machine $\mathcal{M}$ is decidable.*

Proof

By the generalized completeness of first order logic in general and COCOLOG in particular, we know that for any formula P, a proof exists for P and there exists a terminating search for such a proof if P is a consequence of $\Sigma^0_{\mathcal{M}}$. Now for any formula P we start a search for all possible proofs for both P and $\neg P$. One of these two searches will terminate since $\Sigma^0_{\mathcal{M}}$ is a complete axiomatization, i.e., either P or $\neg P$ will be a consequence of $\Sigma^0_{\mathcal{M}}$. Thus we can conclude that the axiomatic theory generated by $\Sigma^0_{\mathcal{M}}$ is decidable.

□

Now if we denote $\Sigma^k_{\mathcal{M}} = \Sigma^0_{\mathcal{M}} \bigcup \Sigma^k$ as the axioms generating the theory $Th(o^k_1)$ together with the size axioms, then the above results for $\Sigma^0_{\mathcal{M}}$ generalize to any logical theory in a COCOLOG family.

5 Extra-Logical Transitions Between Logical Theories

In order for the family of logics in a COCOLOG to work coherently, certain requirements have to be met. These requirements can be viewed as restrictions on the transitions between logical theories which cannot be represented within these theories themselves. As a result, the extra-logical features of the transitions are governed by meta-level assumptions including a meta-level rule of inference.

The meta-level axioms assumptions are that there are no errors on the observation and control channel and the control commands sent from the logic controller will be implemented *instantly* and *correctly*. Hence there will not exist any conflict between observation and control axioms and reality as represented by the mathematical model $\mathcal{M}$. Further we assume that the entire deductive closure $Th(o^k_1)$ of the axioms Σ^0_k is instantaneously generated in the logic regulator $\mathcal{R}$ of a COCOLOG feedback system at each discrete time instant k.

From the definition of Σ^0_k in Section 4 we evidently have the following rule of inference which connects theories at different instants along a trajectory of observations and control actions for a finite machine.

Nesting of Theories (Meta-Level Rule of Inference)

$A \in Th^o(o^k_1)$ implies $A \in Th^o(o^{k'}_1)$, for any $k' > k$ and so the sequence of theories satisfies the following condition

$$\cdots \subseteq Th^o(o^k_1) \subseteq Th^o(o^{k+1}_1) \subseteq \cdots$$

□

We see that this sequence of COCOLOGs combined with the meta-level requirements constitute a closed loop feedback logical control system as displayed in Figure 5.1

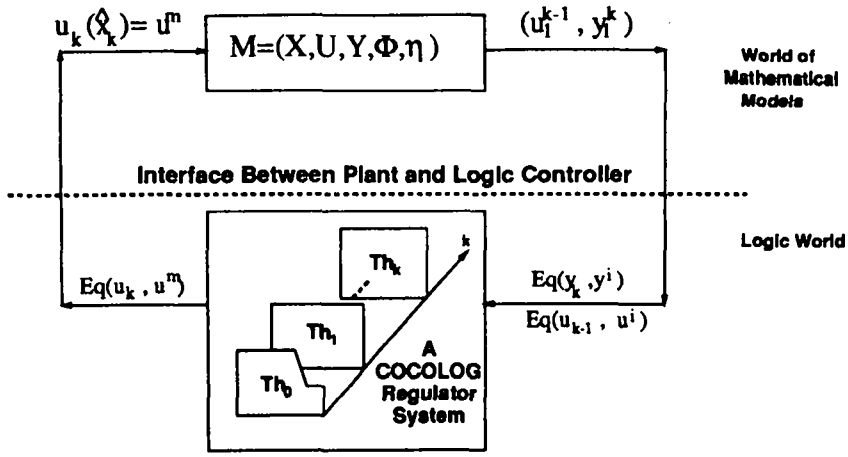


Figure 3: A Closed Loop Logic Control System

6 Conclusion

Many questions concerning COCOLOG may be posed at this point. Perhaps the most important of these are the following:

(i) The definition of a tractable, analyzable, fragment of COCOLOG obtained by suitably restricting the class of CCAs.

(ii) The issue of implementability of COCOLOG for real time systems leads one to the question of automatic theorem proving in COCOLOG. As remarked in the Introduction, current experiments using the FE-resolution extension of the GTP automatic theorem proving software of Newborn [1987] (developed by Q-X. Yu) are encouraging (see Wang and Caines [1992]). A complexity analysis of such algorithms would be most valuable.

(iii) A realization of a COCOLOG is a sequence of first order theories generated by a given sequence of input-output observations and it corresponds to a path in the COCOLOG tree structure (see Figure 3.1). The true formulas in the nodes of this tree can be captured by a possible world interpretation of a modal logic, see Goldblatt [1987]. Instead of modal logic however, in the current analysis we have used a family of classical first order logics to codify the state observation and control problem. We believe a modal logic representation would be too restrictive. The word restrictive is used in the following two senses: First, it most easily represents a static world. In other words, a modal logic cannot handle in a simple manner unknowns or the changes in the dynamics, or the environment, of a system and this prohibits its use as a logic for real time control tasks. Second, it is not necessary to code all the paths of an observation tree into a control logic since a physical system cannot realize all such possibilities. Therefore the extra coding of modal logic system would tend to delay its response time. Despite these reservations, a study of the mathematical properties of an overall modal logic formulation of COCOLOG families of theories merits attention.

Acknowledgements

The authors gratefully acknowledge conversations concerning this paper with Tom Mack-

ling, Michael Makkai and Yuan-jun Wei and to David Delchamps for the name COCOLOG.

References

P. E. Caines and S. Wang. Classical and Logic-Based Regulators for Partially Observed Automata: Dynamic Programming Formulation. In *Proceeding of the 1989 Conference on Information Sciences and Systems*, John Hopkins University, Baltimore, MA, March 1989a.

P.E. Caines and S. Wang. Classical and Logic-Based Regulator Design and Its Complexity. In *Proceeding of the 28th IEEE Conference on Decision and Control*, Tampa, Florida, December 1989b.

P.E. Caines and S. Wang. "COCOLOG: A Conditional Controller and Observer Logic for Finite Machines", *Proceedings of the 29th IEEE Conference on Decision and Control*, Hawaii, 1990.

P.E. Caines, R. Greiner, and S. Wang. Classical and Logic-Based Dynamic Observers for Finite Automata. *IMA Journal of Mathematical Control & Information*, Vol.8, pp. 45-80, 1991.

P.E. Caines, S. Wang, and R. Greiner. Dynamical Logic Observers for Finite Automata. In *Proceeding of the 1988 Conference on Information Sciences and Systems*, pp. 50–56, Princeton University, Princeton NJ, March 1988.

R. Goldblatt. *Logics of Time and Computation.* CSLI/Stanford, Stanford, CA, 1987.

W. Kohn, A Declarative Theory for Rational Controllers. *Proc. of the 27th IEEE CDC*, Vol.1,pp.131-136, Austin, Tex., December 7-9, 1988.

W.Kohn, Declarative Control, *Communications of the ACM*, Vol.34, No.8, pp. 65-79 August 1991.

F. Lin and W.M. Wonham, On the Observability of Discrete-event Systems, *Inform. Sci.*, Vol.44, No.3, pp. 173-198, 1988

E. Mendelson. *Introduction to Mathematical Logic.* Van Nostrand Reinhold Company, New York, N.Y. 10001, 1964.

M. Newborn. *The Great Theorem Prover*, Newborn Software, P.O.Box 429, Victoria Station, Westmount, PQ, Canada, 1989.

J.S. Ostroff. *Real-Time Computer Control of Discrete Systems Modeled by Extended Machines: A Temporal Logic Approach.* PhD thesis, University of Toronto, Jan 1987.

J.S. Ostroff. *Real Time Temporal Logic*, John Wiley, NYC, 1989a.

J.S. Ostroff. Synthesis of Controllers for Real-time Discrete Event Sytems. Technical Report CS-98-09, Department of Computer Science, York University, June 1989b.

J. Ostroff and M. Wonham. A temporal logic approach to real time control. In *Proceedings of 24th IEEE Conference on Decision and Control*, 1985.

J.S. Ostroff and W.M. Wonham. A Framework for Real-time Discrete Event Control. Systems Control Group Report 9810, Department of Electrical Engineering, University of Toronto, July 1989.

S.J. Rosenschein and L.P. Kaelbling. The Synthesis of Digital Machines with Provable Epistemic Properties. Technical Note 412, SRI International, April 1987.

P. Ramadge and W.M. Wonham. Supervisory control of a class of discrete-event processes. *SIAM J. of Contr. Optimization,*, 25(1), Jan 1987.

P.J. Ramadge and W.M. Wonham. The Control of Discrete Event Systems. In *Proc. IEEE, Vol.77, No.1*, pages 81–98, January 1989.

J.G. Thistle and W.M. Wonham. Control problems in a temporal logic framework. *International Journal of Control*, 44(4), 1986.

M. Wonham and P.J. Ramadge. On the supremal controllable sublanguage of given language. *SIAM J. of Contr. Optimization,*, 25(3), May 1987.

S. Wang. *Classical and Logic Based Control Theory for Finite State Machines* Ph.D. Thesis McGill University, Montreal, October, 1991.

S. Wang and P. E. Caines. *Automated Reasoning with Function Evaluation for COCOLOG* Proceedings of pre-MTNS Conference, Hangzhou, P.R. China, To appear: World Scientific, Singapore, 1992.

ISSN 0249-6399